Z-80c

User's Manual

APPLIED ENGINEERING
P.O. Box 798, Carrollton, Texas 75006

CONGRATULATIONS!   You   now own APPLIED ENGINEERING'S Z-80   card.
Applied   Engineering   is   a   leading   manufacturer   of   Apple
peripherals.   Because   of the time and care taken in the   design
and   manufacture   of your Z-80 card,   we are sure that   you   will
enjoy the use of it for many years to come.

This   manual was written with a Z-80c and Wordstar using an Apple
IIc.

2.0

Table of Contents

INSTALLING THE Z-80c IN YOUR APPLE

The Z-80c card simply plugs into a connector inside your Apple. Care must be exercised however, so follow these instructions exactly.

Applied Engineering has conducted very extensive testing with the installation procedures outlined below. However, Applied Engineering will assume no responsibility should you damage your IIc cabinet or electronics. If you follow the procedure outlined below, your IIc will come apart with little or no difficulty and it is virtually impossible to install the Z-80c incorrectly. Even if you did, you would not damage the IIc circuitry.

At the present time, it is not known whether you'll void your Apple IIc warranty by opening the case. However, if your IIc should need warranty work the Z-80c can be removed from your computer and it will not be possible for anyone to know that you were ever inside.

To prepare your IIc, please unplug all cables from it including the power cord, video cord, mouse, printer and second disk drive. The tools required are a medium sized Phillips screwdriver and a medium sized flat blade screwdriver.

We recommend that you perform the installation on a soft surface so you will not mar the finish of the cabinet.

A step-by-step procedure follows:

In the unlikely event that you encounter difficulties in installing the Z-80c in your IIc, please read the section in the manual entitled "In case of difficulties" before you call Applied Engineering for assistance as most problems have been anticipated and are discussed in this section.
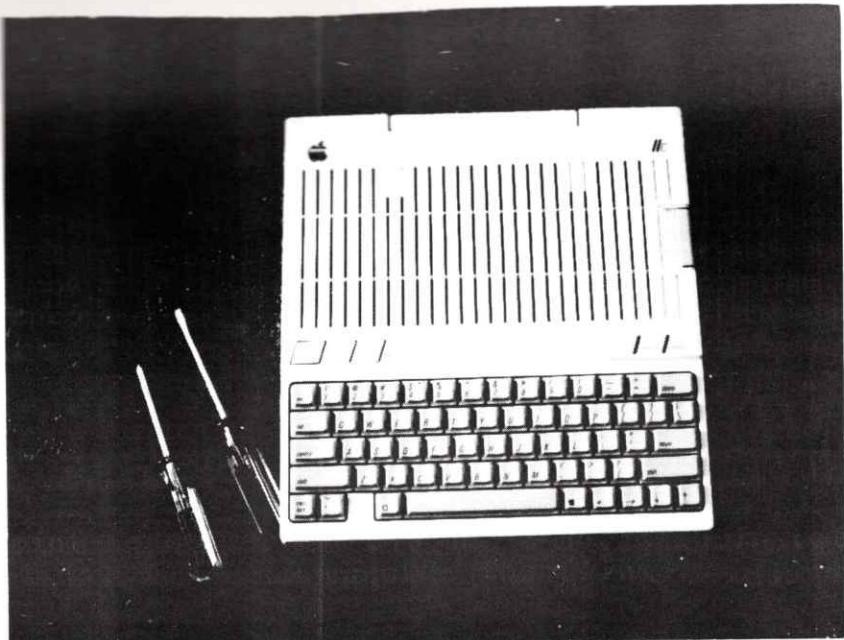
Everything you need for installation. Your IIc, Phillips
screwdriver and flat blade screwdriver.
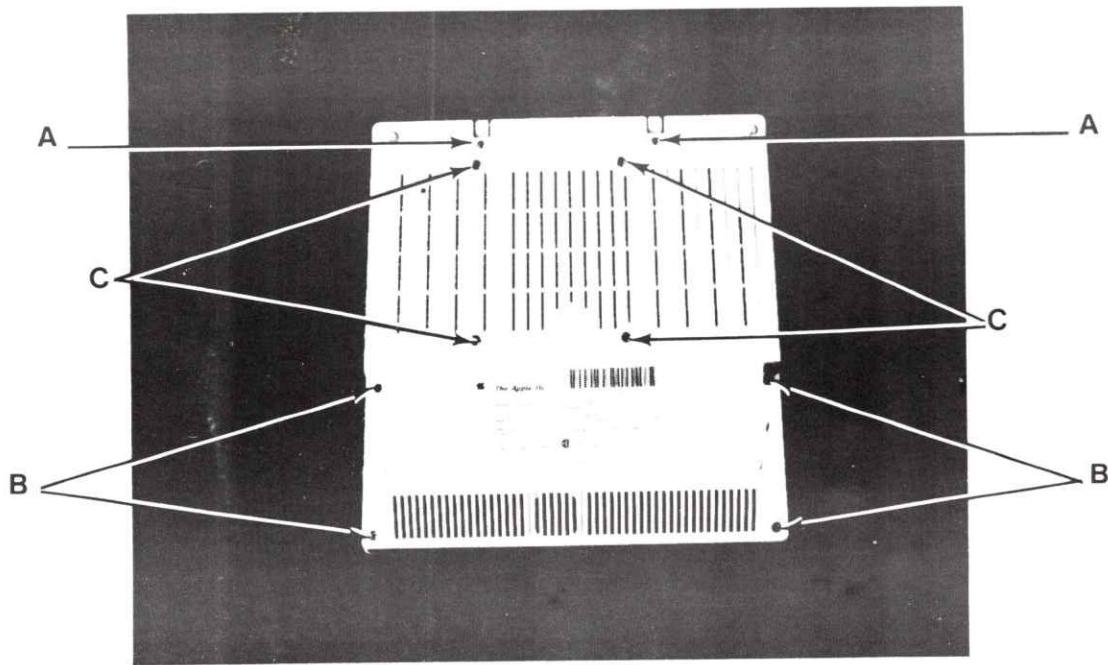
Turn the IIc upside down and using the Phillips screwdriver,
remove the screws indicated as A, B and C. The A, B and C screws
are of different thread types so keep them separated so that you
can reinstall them in their original positions as shown. Some of
the screws may not unscrew completely, this is normal.
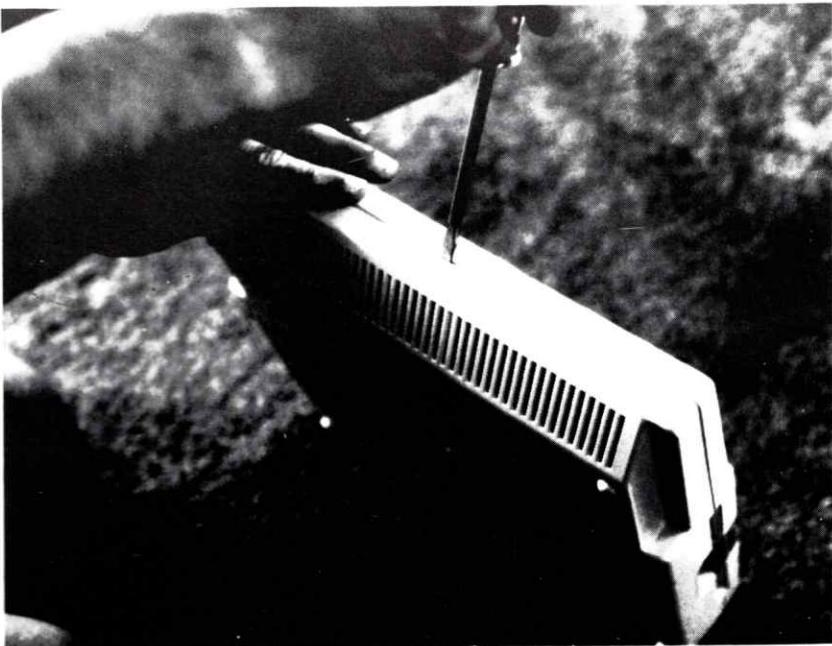
There is a small plastic snap that holds the cover onto the IIc. This snap may be released by pushing down on the top cover until it becomes free. If it is stubborn, you may want to use the flat bladed screwdriver to pry it loose. Also, refer to the photograph below.
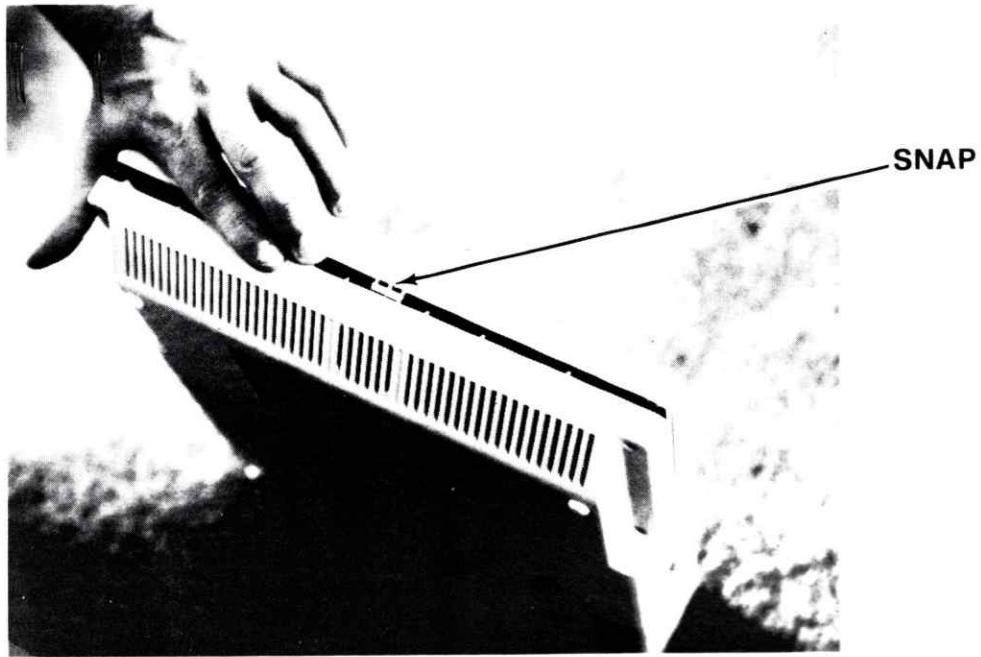
SNAP

Should the snap break off, it is of little consequence as it is only used by Apple so that they can put the IIc together a few seconds faster. When you reassemble the IIc, the screws on the back hold it together firmly.
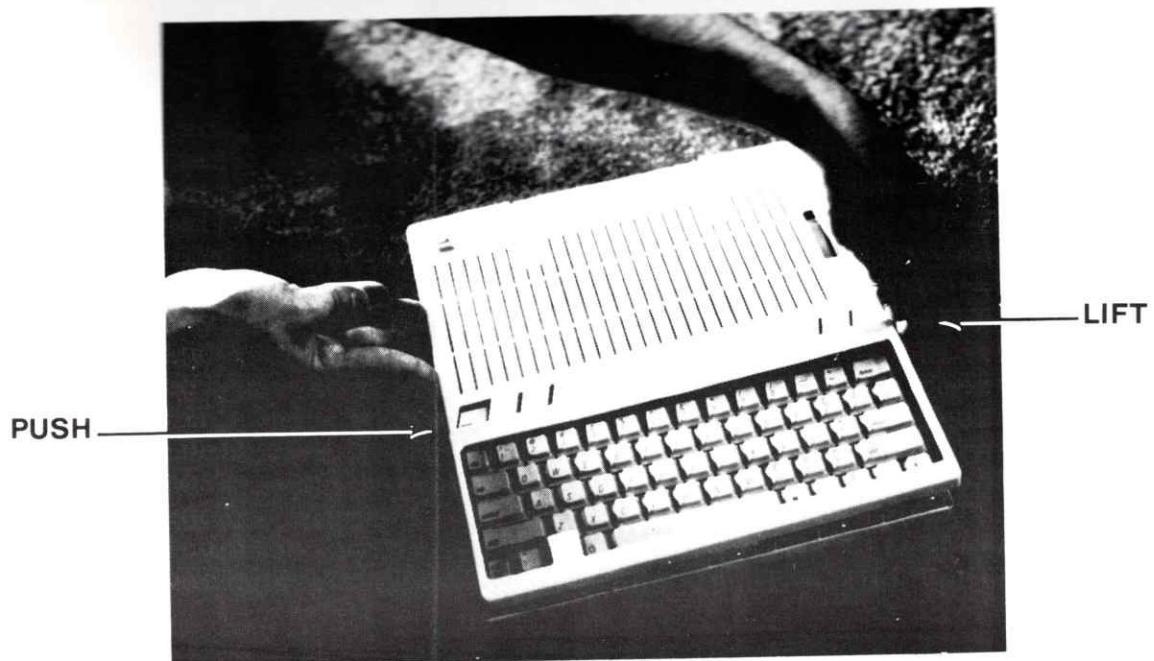
PUSH

LIFT

Push on the top cover (the side with the Apple logo) and lift on
the other side (the disk drive side).   At the same time, loosen
the back panel which is a separate piece. On some IIc's this may
require a bit of wiggling, but it will come off.   If you break
any snaps, don't worry as there are twice as many as necessary.

DISK DRIVE

POWER
SUPPLY
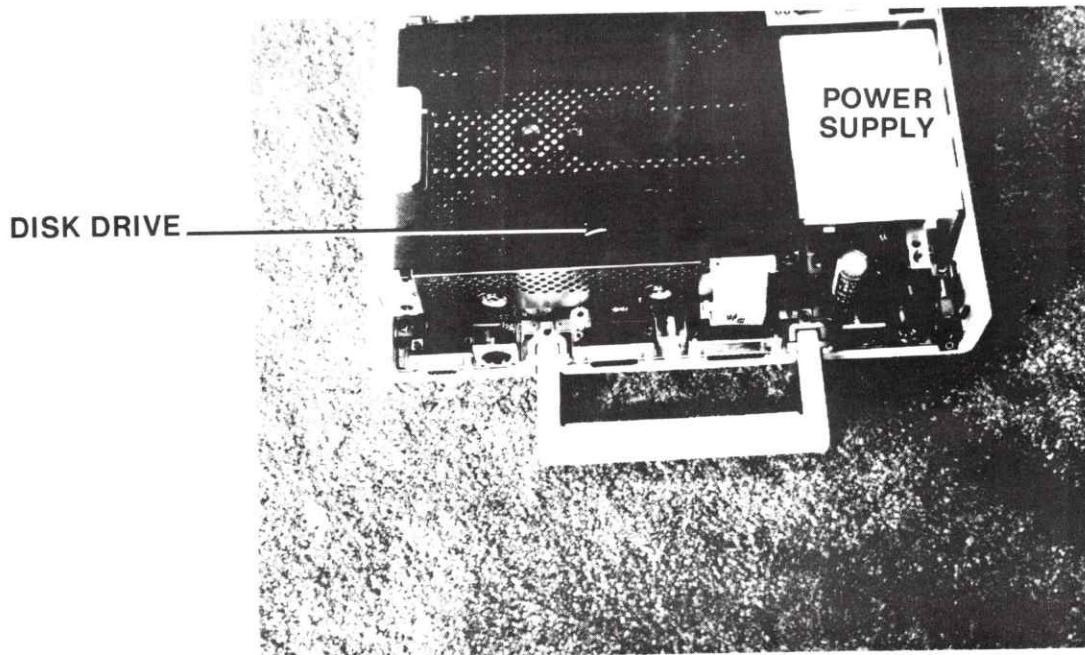
Now try to lift out the disk drive.   If it won't come out, you
may need to loosen the screws that hold it in some more.
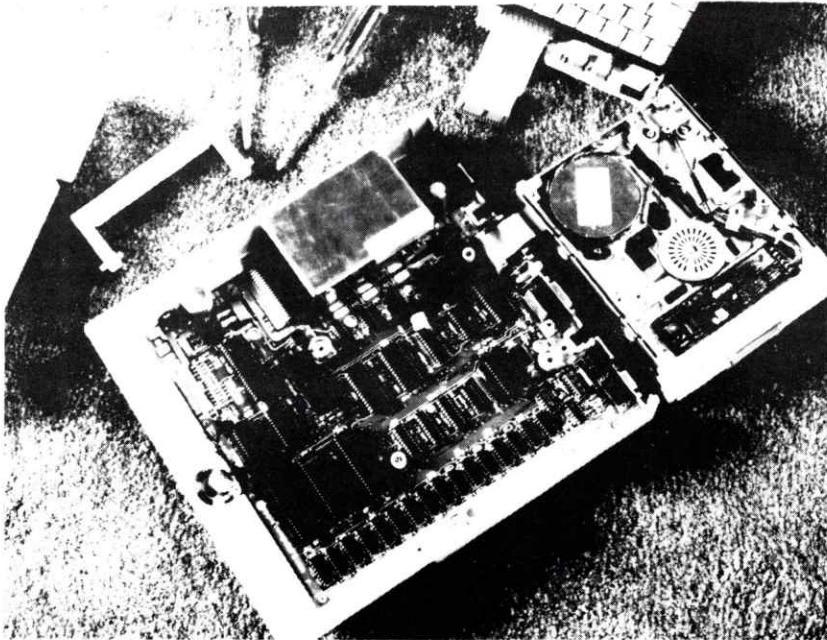
The disk drive and the keyboard will come loose pretty much at the same time. Now, unplug the flat cable that connects the keyboard to the IIc motherboard and set the keyboard aside. Now go get the Z-80c.

NOTE
NOTCH
ON CHIPS
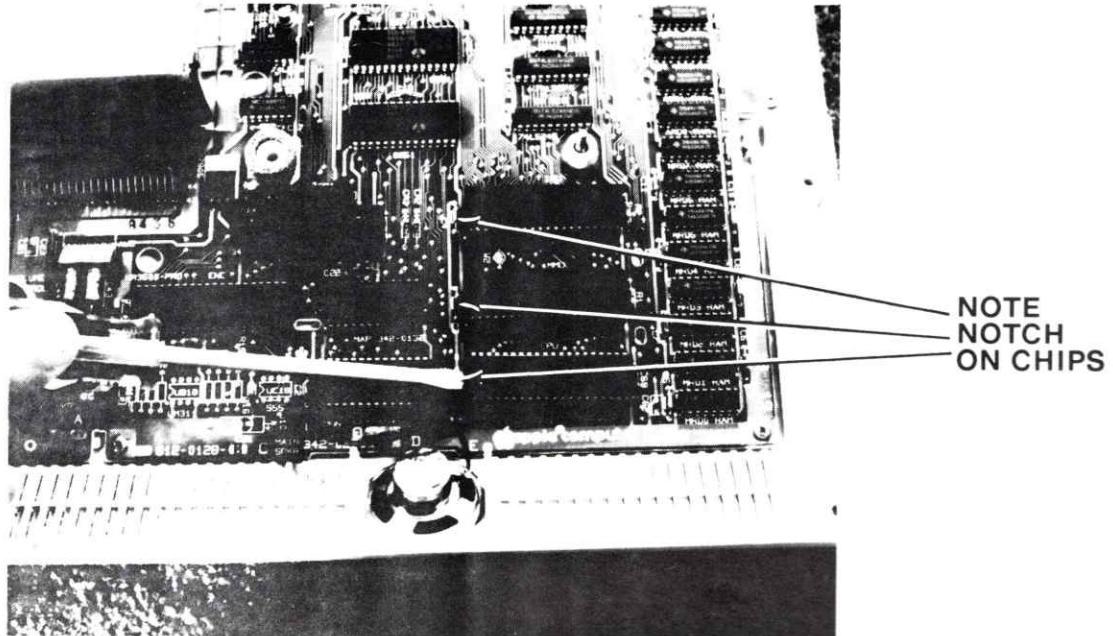
Touch the Apple power supply (large metal box). Using a flat bladed screwdriver, unplug the 65C02 chip and plug it into the empty socket on the Z-80c. The notch on the 65C02 is oriented toward the edge of the Z-80c. This notch points away from where the wire attaches to the Z-80c card.

TMG

C13
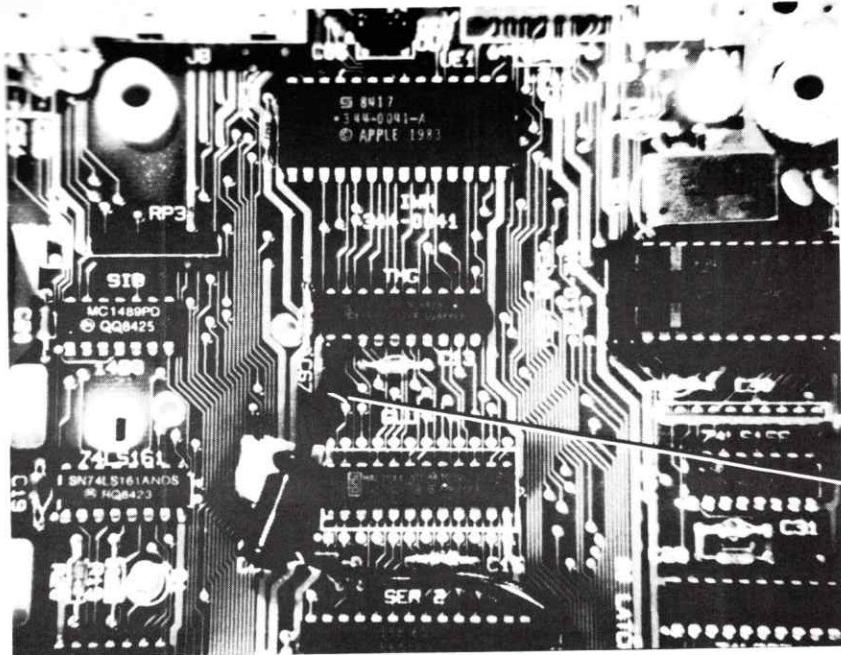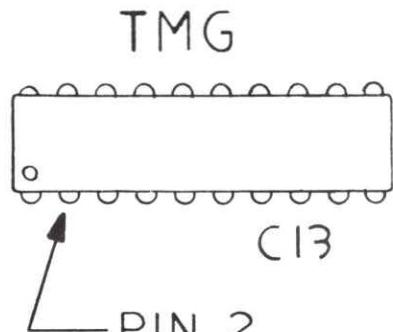
PIN 2

GRABBER

There is a short wire connected to the Z-80c card. On the end of
this wire is a small plastic "grabber". When you press the
grabber together, a metal hook will project out of the housing.
Hook the grabber to pin 2 on the "TMG" I.C. Use the small piece
of double sticky tape provided to hold the grabber down flat.
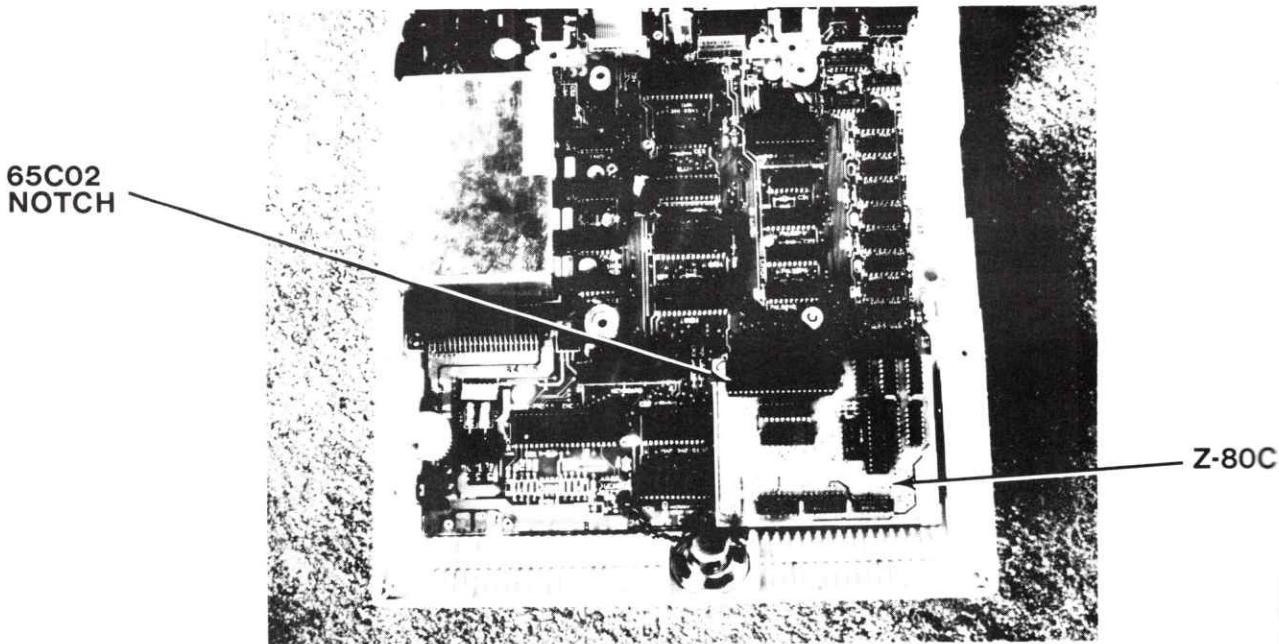
65C02
NOTCH

Z-80C

Touch the Apple power supply (large metal box). On the underside
of the Z-80c, you will notice a 40 pin connector. This connector
plugs into the socket where your 65C02 was. Plug the Z-80c into
this socket, paying special attention so that all 40 pins go into
the socket. If you're not careful you can install the card one
pin too far to the right or to the left.
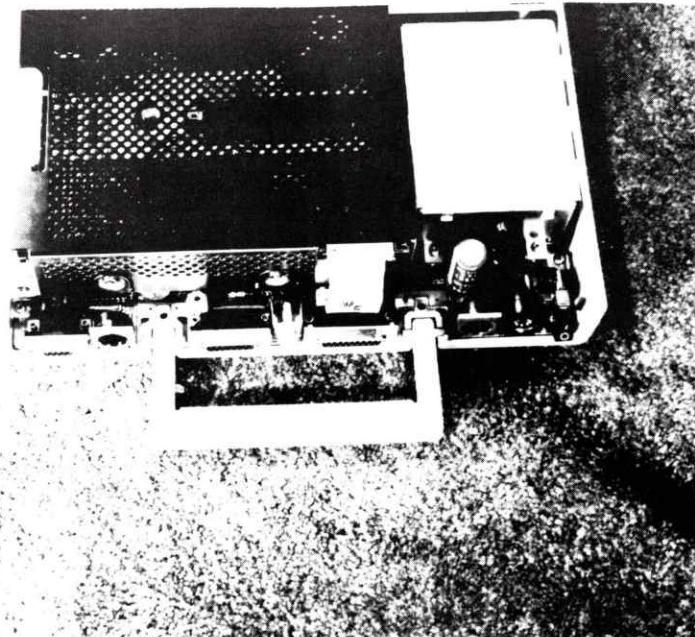
Reinstall the disk drive and plug the flat cable from your
keyboard back into your motherboard (it will only go one way).
At this time you may wish to carefully move your IIc back to your
work station and temporarily connect the power cord and monitor
only. In order to test your installation, insert a DOS 3.3 or
ProDOS system master diskette and turn on your IIc. The disk
should boot as normal. If it does not, turn the computer off and
refer to the section "In case of difficulty". To test the CP/M
operation, insert the disk that came with the Z-80c and see if
it boots. If all goes well, turn the IIc off and unplug the
power cord and monitor cord and carefully carry it back to where
you were.
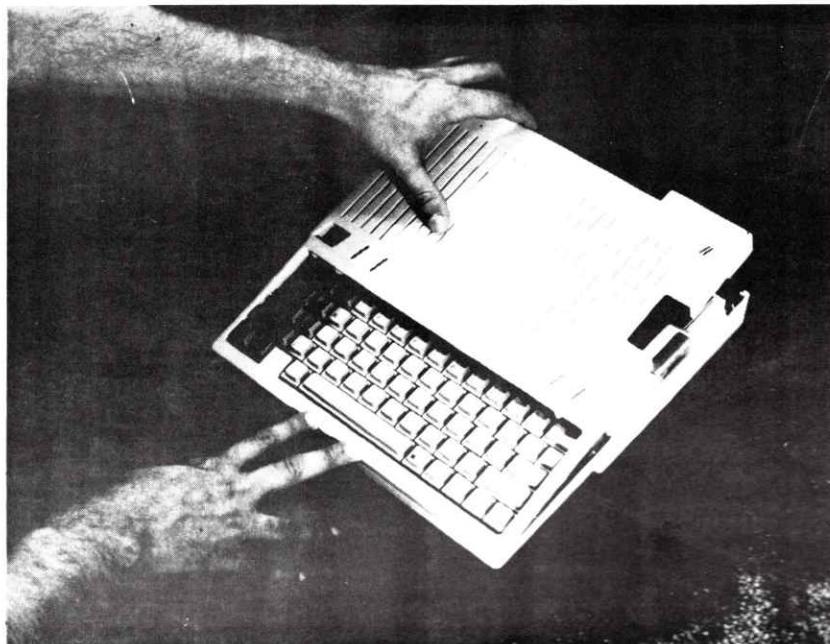
Reinstall the cover by hooking the front plastic snap first and
tilting it downward.
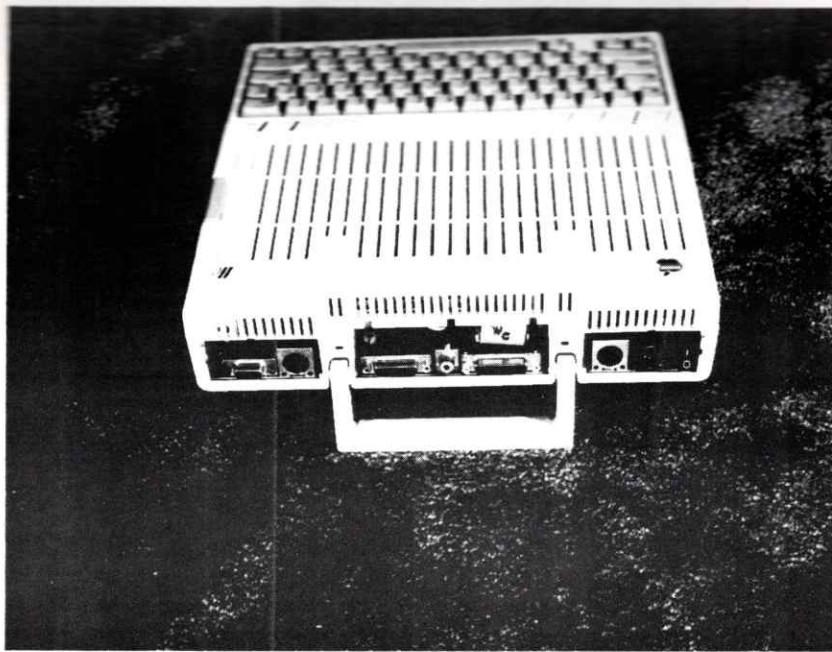
Make sure you've got the carrying handle in position.
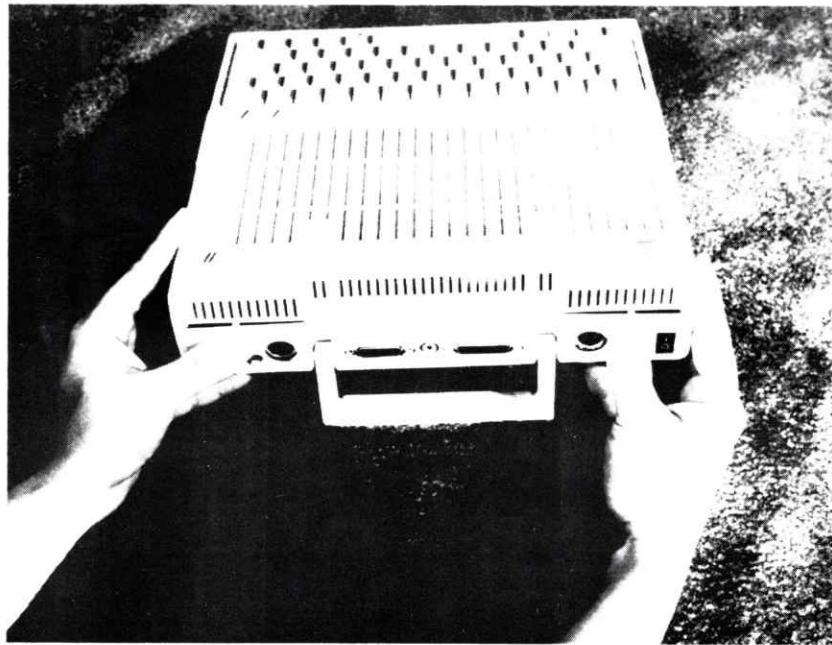
Snap the back panel back into place and you're all done.

Once the Z-80 card is installed in the computer, you're ready to start using CP/M software. The CP/AM 4.0B system master diskette that came with your Z-80 card should be copied using the following procedure.

1)   Insert CP/AM master diskette in drive 1 and turn on computer.

2)   The copyright notice will appear in the upper left hand corner of the screen. Just below the copyright notice will be the CP/AM prompt 'A>'. At this point, type the following:

COPY B:=A:

If your system has only one disk drive, then type:

COPY A:=A:

Following the directions on the screen, place a blank diskette in drive B (slot 6, drive 2) for a two drive system, whereas single drive users will have to swap disks at the designated time.

3)   Press return key to begin copying.

4)   Then make a second backup copy.

5)   Label the two backup diskettes and store the original in a safe place.

As a general rule, original diskettes should never actually be used, but rather only be used to make copies from and then use the copies.

Larger Memory Capacity

If the system being used is an Apple IIe, Apple //c, or an Apple II or II+ with a 16K language the CP/AM operating system can be upgraded to a 60K system from the standard 44K system that is already on the disk. This can be done by executing the CPAM60 program on each of the backup diskette as shown below:

A>CPAM60

and following the instructions. Additional information about this program can be found in the transient program section.

Pressing RESET

DON'T PRESS RESET - When programming in Apple BASIC, some people have gotten used to pressing Reset to break out of a program. This habit must be broken when using CPM software. Remember, official Apple doctrine states that when Reset is pushed "all bets are off".

SOFTWARE COMPATIBILITY

CP/AM 4.0B is compatible with virtually all older CPM systems. When purchasing software, you need not look for any mention of CP/AM 4.0B specifically as this operating system is compatible down to CPM 1.6. 99.9% of all the programs that you will buy will require CPM 2.2 for which CP/AM 4.0B is compatible.

CREATING A SYSTEM DISKETTE

A system diskette is one that will load and initialize CP/AM when booted. To create a sytem diskette on a two drive system, type:

COPY B:=A:/S

On a one drive system type:

COPY A:=A:/S

The /S means copy only the system and do not copy the other files on the disk. The new CP/AM system disk will now boot and can be used to store programs and data. The /S command is somewhat similar to the DOS 3.3 INITialize except that /S will not erase programs on the diskette. This command is also used to put the operating system on commercially prepared programs so that they will cold boot with no preboot. Many commercially prepared programs will refer to a SYSGEN command. The /S is the SYSGEN under CP/AM 4.0B. Only use the /S command on BACKUP copies of programs because if you should make an error, the original disk could be damaged.

INTRODUCTION TO CP/AM

FILE NAMES

File names in the CP/AM system follow the following format:

        d:filename.typ

The drive specification, which is optional, identifies which
drive the file is located on is given by 'd:', where the lower
case 'd' can be any drive name. The 'filename' can be from 1 to 8
characters, this can be followed by an optional file type
specifier. The type specifier of the filename must be preceded
by a period followed by a maximum of three characters. The
period is not part of the file name, but only a separator. There
are a few characters that are not allowed in a file name and are
listed below:

        < > . , ; : = ? * [ ]

There is a general convention that is followed by the CP/M world
on what special file type names mean. These file type names
identify the format of a file, which is used by many programs.
To avoid problems, it is recommended that the following type
names be used only for their intended purpose:

        .ASM   Assembly language source code for the ASM program.
        .MAC   Macro assembly language
        .HEX   Hexadecimal file output by assemblers
        .REL   Relocatable file output by assemblers
        .COM   Command file that can be executed by entering its
                    file name alone
        .PRN   Print file
        .LIB   Library file of programs
        .SUB   Input for CP/M type batch facilities
        .C   C language source code
        .PAS   Pascal source code
        .COB   COBOL source code
        .FTN   FORTRAN source code
        .TXT   Text files
        .DOC   Documentation files
        .$$$ Temporary files

Through the use of special 'wild card' characters CP/AM has the
capability to refer to one or more files at the same time.
There are two special characters that can be used, the '?' and
'*', of these the '?' is the main wild card character. Where
ever the '?' character appears in a file name that position will
be matched with any character. Using the '?' character all of
the files on a disk can be matched by typing '????????.???'. As
an example, the files 'TEMP1.DOC' and 'TEMP2.DOC' can be matched
at the same time with 'TEMP?.DOC'. Although the file name
'TEMP10.DOC' would not match because it has another character

13

after the position that the '?' matches. The file name 'TEMP??.DOC' would however match all three file names.

To keep from having to type in many '?' characters another wild card character is provided, the '*'. This means to use as many '?' as needed to fill out the rest of the name or type field. Thus, '????????.???' can be written as '*.*' and 'TEMP10.DOC' can be matched with 'TEMP*.DOC'.

Default Disk Drive

When a filename is used without a drive specifier, the default drive will be assumed for the operation. The default drive is indicated at the CP/AM prompt by the letter just to the left of the greater than sign. When the CP/AM system is first booted up the prompt 'A>' will be displayed. This shows that the default drive is 'A' and that the system is ready for a command. In the following command line example:

B>A:STAT TEMP.TXT

the default drive is 'B'. The transient program STAT has a drive specifier of 'A:' so the system will look on the disk that is in drive 'A' for the STAT program and if found, will execute it. The filename parameter of STAT is 'TEMP.TXT' which has no drive specifier. When the STAT program begins to run it will look for the file 'TEMP.TXT' on the default drive 'B' and if found, will display the file size.

To change the default disk drive, simply type the drive letter followed by a colon, then return. For example, to go to drive B just type 'B:'. If the disk drive is not currently logged in, the CP/AM system will read the directory information from the disk and log it in. After this, the CP/AM prompt 'B>' will appear. If the disk is already logged in, the system will not have to read the directory and the prompt will appear immediately. From this point on, all further disk commands will use drive B as the default.

Each disk drive is assigned a letter name followed by a colon as shown below:

CP/AM name		Slot Number	Drive Number
1st Drive	A:	6	1
2nd Drive	B:	6	2
3rd Drive	C:	5	1
4th Drive	D:	5	2

Changing CP/AM Disks

Unlike Apple DOS, you cannot indiscriminately change disks in the drives with CP/AM. When you change disks, you must let CP/AM

know that you have done so. This is because there is certain
information stored in memory about the disk in use. When you
change disks, this information should be replaced with the
information of the newly inserted disk. To let CP/AM know that
you've changed disks, type control C after you've changed disks.
If you do not type control C after changing disks and attempt to
write to the changed disks, the computer will display:

BDOS ERR ON d:

If this message is displayed, press control C to return to CP/AM.
Control C may be typed anytime you think it may be needed for it
will not hurt anything.

There are several ways to get the CP/AM operating system started
in the Apple. The easiest way is to simply turn on the computer
with the CP/AM disk in drive A. Another method is to type
CONTROL, OPEN APPLE, RESET (Apple IIe only). This is done by
pressing the Control key and Open Apple key down simultaneously
with the left hand, then pressing Reset with the right hand. If
Apple DOS (3.3 or Pro-DOS) is currently running type PR#6 and
press return.

Once CP/AM has been booted, you are ready to begin experimenting
with its various commands. If the disk will not boot, please see
the section on page 30 called 'IN CASE OF DIFFICULTY'.

CP/AM Commands

CPM commands fall into two categories, system commands and
transient commands. System commands are built into the CP/AM
system and always available whereas transient commands are
programs that are called up from the disk drives. If you ever
try to execute a transient command and the computer responds with
a question mark, this is because that particular command is not
on the disk that you're using. It is a simple matter to copy
these transient commands to other disks and this procedure will
be outlined later.

CP/AM System Commands

DIR

DIR stands for Directory and is similar to typing Catalog under
Apple DOS. To use this command, simply type DIR at the CP/AM
prompt 'A>' followed by pressing the return key and it will give
a listing of all the files on the currently logged on disk.

```
A>DIR
A: PIP      COM  28 : CPAM60   COM  42 : RESTOR  DOC    5
```

In the sample Directory above, three files are shown. They are

15

PIP.COM, CPAM60.COM and RESTOR.DOC. One of the unique features of CP/AM 4.0B are the numbers following each file name. These numbers indicate the disk space that each file uses in pages. One 1K (1024 bytes) of memory equals four pages, so the PIP file is 28/4 or 7K bytes long.

By typing DIR and getting a successful directory, the Z-80 card and the computer have been given a thorough hardware test. Should you ever encounter difficulties in the future, the system operation can be tested by cold booting on a master diskette and typing DIR. If this doesn't give successful directory, then the problem could be hardware related. If a proper directory is displayed, then contact the software publisher for help.

To get a directory of a different drive, just type 'DIR B:' for the directory on drive B. After displaying this directory, the system will be return to drive A. However, don't try to get a directory on drive B unless there is a CP/AM disk in that drive.

Another way of displaying the directory is to use the transiant command SDIR. SDIR will sort the directory into alphabetical order and display some additional information about the disk. For more information refer to the transiant command section to the manual.

ERA

ERA is the erase command. Like DIR, it is a system command that can be used on any disk because it is built into the CP/AM operating system. Its format is very simple:

A>ERA B:TEMP.BAK

The above command will erase a temporary file on drive B:. The ERA command is somewhat unique because the drive letter must be specified and is not optional as in other CPM commands. To erase all the files on a disk (drive A: in this example) type :

A>ERA A:*.* A:*.*

When the return key is pressed all of the files on the disk will be removed. Be sure that this is really what you want to do for there will be no question asked before the ERA command starts.

REN

This command allows you to change the name of a file. The format is:

REN new.typ=old.typ

For example, assume the FILE21.TXT is on the diskette in drive A. After typing:

```
A>REN FILE33.TXT=FILE21.TXT
```

FILE21 will be renamed to FILE33, and only FILE33 will be on the
diskette in drive A.   The name FILE21 will have been discarded.
The file itself, however, will remain intact.

TYPE

The TYPE command can be used to display any ASCII file on the
screen.   This is a quick and convenient way to examine any
textfile. The format is:

```
A>TYPE d:filename.typ
```

The TYPE command can be used to view any text files on a disk.
While it will not provide the convenient format of a word
processor, TYPE does print the file as it is actually stored on
the disk.   It is a common practice to store documentation for
many programs on a CPM disk as a filename.DOC or filename.TXT.
For example, on the CP/AM master diskette will be found a program
called 'NSWEEP.DOC'. After typing the following:

```
TYPE A:NSWEEP.DOC
```

the latest documentation on the program 'NSWEEP.COM' will be
displayed.   The TYPE command will display a page of information
at a time.   To view the next page, simply press return or for
finer control, use control S. The listing can be stopped by
typing control C.

USER

If several people or application programs share a single disk,
the user command can be used to separate them onto different
parts of the disk directory called USER AREAS.   These user areas
are normaly not used, however it can be very useful on systems
that have hard disk drives.

The disk directory for each disk drive is be divided into sixteen
different user areas.   Each user area is logically separate from
all other user areas except for user area zero, which is global
to all user areas in the CP/AM system.   When in user area zero,
only the files contained in user zero will be displayed by the
DIR command.   All files that may be in the other user areas are
not directly accessible from user zero. When in any of the other
nonzero user areas the files contained in both user zero and
those in the current nonzero user area are accessible and will be
displayed by the DIR command.   This allows different sets of
related files to be isolated in thier own user areas with the
common system programs that may be used by several users areas,
such as text editors and compilers, to be placed in user area
zero and shared between all user areas.   In most circumstances,

as mentioned above, the user area facility can be ignored and all
work done in user zero. If user areas are going to be used, care
must be taken so that no filename is created in a nonzero user
area that is also used in user area zero.

To move to a different user area type 'USER' followed by a number
from zero to fifteen.   The example below will move to user area
one:

A>USER 1

The CP/AM prompt will change to:

aA>

this indicates that user area one is now active.   The lower case
letter to the left of the default drive letter will be 'a' for
user 1, 'b' for user 2,... etc. through user fifteen.   The DIR
command will indicate which files belong to user zero and which
belong to the current user area by prefixing the files contained
in the nonzero user area with a lower case letter of that user
area. This is shown in the example below:

A>USER 1
aA>DIR
A: PIP   COM 26: STAT   COM 18:aLETTER   TXT   8:
aA>

Both of the files PIP and STAT belong to user zero and the file
LETTER.TXT belongs to user one.

SAVE

The save command allows a copy of the system memory to be saved
to a file on disk. The format for the SAVE command is as follows:

        SAVE n filename.typ

where n is the number of 256 byte pages to be saved starting at
hex address 100H.   The following example will save 2048 bytes of
the system memory starting at 100H to the file 'MEMSAVE.XXX':

A>SAVE 8 MEMSAVE.XXX

PIP

It is often desirable to move some of the utility programs from the system master diskette to a program diskette. To do this, a method of transferring files from one diskette to another is needed.

The PIP program is used to perform copy operations. It is a program that you execute by typing 'PIP'. With the system master diskette in drive A and one of the diskettes that you recently formatted in drive B, type 'PIP'.

```
A>PIP
*
```

The asterick prompt is the PIP program prompt and indicates that PIP is up and running. A PIP expression that will copy a file(s) from one disk to another can now be typed. Simple PIP expressions take this form:

```
d:copyname.typ=d:originalname.typ
```

The 'copyname' and 'originalname' arguments are actual filenames, and the 'd' stands for the letter of the drive. The '.typ' means the file type that has been designated for that file. The extension '.typ' for PIP is '.COM'. PIP always copies from an original file to a copy file. Thus, the original file must already exist; PIP creates the copy file with the name you supply (normally the same). For example:

```
*B:COPY.COM=A:COPY.COM
```

This expression tells PIP to make a copy of COPY.COM which is on drive A and to name it COPY.COM and put it on drive B. More files can be copied from this point or, by pressing the return key, return to the CP/AM 'A>' prompt.

The new file can be given a different name during the transfer as follows:

```
*B:OLDCOPY.COM=A:COPY.COM
```

If only one PIP operation is desired, then a shorter commmand can be used by typing the PIP expresssion on the PIP command line:

```
A>PIP B:COPY.COM=A:COPY.COM
```

When PIP stops copying, the system prompt will appear again. If the file name is not going to change, an even shorter method is available.

```
A>PIP B:=A:COPY.COM
```

This command will put a copy of COPY.COM on drive B.   This is
probably the most common CP/AM command of all.

It's easy   to copy all files on a diskette with PIP by making
greater use of the wild card characters:

A>PIP B:=A:*.*

Remember, filenames can have eight characters to the left of the
period, and three characters to the right of the period.   The *
symbol matches any eight characters to the left of the period,
and the .* symbol matches any three characters to the right of
the period.   When using the above command, remember that only
files are copied, the CP/AM system will not be copied.

The PIP command also has the ability to verify if the file(s)
have been transferred correctly by rereading the new file and
comparing it to the original. PIP can be told to verify the file
transfers by adding '[V]' suffix to the end of the PIP expression
as follows:

d:copyname.typ=d:originalname.typ[v]

An exmple of this is:

A>PIP B:COPY.COM=A:COPY.COM[V]

After PIP has transferred the entire file, it will go back and
verify that the two files are exactly the same.

STAT

The STAT command is used to find the general amount of space that
is left on a disk or the size of a file. The STAT program can be
executed by typing 'STAT'.   A standard output from STAT may look
like this:

A>STAT
A: R/W, Space: 34k

This indicates that drive A: can be read and written to and has
34k of free space left. An Apple disk runnig CP/AM 4.0B can store
a total of 126K on a system disk and 138K on a non-system disk.
A system disk is defined as a disk that contains the operating
system on it so that the disk can be used to boot up CP/AM 4.0B.
The STAT command will display the drive information for all the
drives that are currently logged-in, so don't be alarmed if more
lines of information appear.   The output from STAT with nothing
after it can be following two forms for each logged disk:

     d: R/W, SPACE: nnnK
  or
     d: R/O, SPACE: nnnK

The R/O means that the disk in that drive has been changed and the new disk has not been logged in with a control C. The files on the new disk can still be read, but nothing can be written to the disk until it is properly logged in.

The space left on a specific disk can be found by typing the following:

```
A>STAT B:
B:  R/W,  Space:  101k
```

This will only display the information for the specified disk. The form for the STAT command is as follows:

```
STAT d:filename.typ
```

The drive specification is optional. If the STAT command is followed by a file name (wild card options are allowed) the status information about a file along with the information about the disk will be displayed:

```
A>STAT FORMAT.COM
 Recs  Bytes  Pex Acc
   26    4k     1 R/W A:  FORMAT   .COM
A: R/W, Space:    34k
```

The Recs column shows how many 128 byte records long the file is and the Bytes column shows the amount of disk space that the file occupies. If the number of records times eight doesn't equal the Bytes colunm don't be alarmed. Because of the way that data is stored on the disk the file size will be rounded up if needed, but as in this example, a total of 4k is used to store the file on the disk. The Pex colunm shows the number of directory extents that the file uses, more information on this can be found in any one of the CP/M texts listed at the end of the manual. Acc specifies if the file can read and written from, if it had been 'R/O' the file could only be read.

A file can be set to a read only status by the following line:

```
A>STAT FORMAT.COM $R/O
File set to R/O.
```

The file is now set to read only and cannot be written to. To change the file so it can be read from and written to use the following STAT command form:

```
A>STAT FORMAT.COM $R/W
File set to R/W.
```

2.0

BOOT

The BOOT command tells your Apple to fire up the disk drive cold. To execute the BOOT command, type 'BOOT' then press return to run the program. If your favorite space game is in drive 1, get ready to dodge missiles. Of course, if the diskette in drive 1 is a CP/AM disk, then you're going to go right back to CP/AM. On Apple IIe's the control, open Apple, reset is probably the easiest way to start fresh. However, many hackers insist that turning off the computer is the only way to start 'clean'.

CPAM60 - Larger Memory Capacity

A CP/AM 4.0B master diskette comes configured as a 44K system and for 48K Apple systems, this is the largest system that can be used. A small minority of programs however, require more memory i.e. dBase II, Turbo Pascal, and WordStar 3.3x. If you have an Apple IIe, //c, or a Franklin Ace 1000, it is recommended that the CP/AM should be upgraded to 60K.

If the system being used is an Apple II+ with a 16K RAM card in slot 0, it can PROBABLY be upgraded to a 60K system. It is emphasized "probably" with Apple II+ owners because some Apple II+'s and some 16K RAM cards have timing problems that are too great for high speed Z-80 operation. If difficulties are experienced in upgrading to the 60K system with the Apple II+, please read the chapter called 'IN CASE OF DIFFICULTIES'.

When using the following procedure to upgrade the CP/AM disk, it is absolutely essential that only a backup disk be used because if something goes wrong the disk may no longer boot up. Also, once the master diskette is converted to 60K, there is NO GOING BACK to 44K, so always keep copies of the original 44K disk around for testing purposes.

To upgrade to a 60K disk type the following:

A>CPAM60

RAMDRIVE - Disk Emulation

When running CP/AM with an Apple IIe, there is a program that will allow CP/AM to use memory in the Apple IIe auxiliary slot as a solid state disk. Disk emulation, simply put, is the act of fooling the Apple computer into thinking that a certain area of memory is a disk drive. The program that does this is called RAMDRIVE. When used with the Apple extended 80 column card, it will emulate half of a disk drive. When used with a 128K or larger RAMWORKS card, it will emulate a full drive or more. To invoke disk emulation, type:

A>RAMDRIVE

Complete information about the RAMDRIVE program can be found in the file 'RAMDRIVE.DOC' on the disk. To view the RAMDRIVE document file type:

A>TYPE RAMDRIVE.DOC

with the system disk in drive A.

RESTOR

The RESTOR program will unerase any file if it is used before any writing is done to the disk. This can be a lifesaver if a very important file is accidently erased. The program is executed with a command of the following form:

A>RESTOR filename.typ

where filename.typ is the name of the file that was erased. If the file could be found, its name will appear on the directory. The file should checked immediately to see if it is all still intact.

AUTORUN

The AUTORUN program will make the CP/AM boot disk automatically run any program when it is cold booted. As an example, this would give the capability to have the RAMDRIVE program automatically install the ramdisk drive when the system is booted. The form of the program is as follows:

A>AUTORUN RAMDRIVE

When the system is cold booted, the RAMDRIVE program will be run just as if it had been typed at the CP/AM prompt. In fact, anything that is typed after that AUTORUN name is taken literally as if it was entered at the CP/AM prompt. To replace any previous AUTORUN command, just re-run the AUTORUN program with the new command. To stop CP/AM from executing any command at boot time, don't enter anything after the AUTORUN name. This program can only modify a system disk ,with CP/AM on it, if it is not a system disk an error message will appear.

FORMAT

FORMAT will completely erase all of the data on a disk and format it in the same way that the INIT command will in DOS 3.3. In order to create a non-system disk that can hold 138k, run the FORMAT program to erase everything (including the CP/AM system) from the disk. As long as the CP/AM system is not put on the disk, using the COPY program, the 12k of storage that was used for the operating system can now be used for data storage. To

run the FORMAT program, type FORMAT at the CP/AM prompt and follow the instructions.

NSWEEP

NSWEEP is a public domain program that has been included on the CP/AM disk. This program is an advanced file transferring program with many extra capabilities. The documentation for this program is in a disk file labeled 'NSWEEP.DOC'. It can be viewed with the type command.

TED

TED is a public domain text editor program that has been included on the CP/AM disk. It is a standard type line editor, like ED, only with online help screens. A complete document file is provided in a squeezed format, which can be viewed or printed out with NSWEEP.

SDIR

SDIR is a public doman program that is very similar to the DIR command that is built into the CP/AM system. SDIR will display all of the files on the disk in alphabetical order and its size in K bytes. After displaying the information on each file it will then display the number of files on the disk and the space used by these files, followed by the space remaining on the disk.

CPORT

This program will allow the default configuration to the Apple //c ports to be changed. Any changes that are made to the defaults will be permanent until the system is powered down, the open Apple reset will not effect the settings. The port configurations can be modified by placing one or more arguments separated by spaces on the command line, as follows:

A>CPORT 5BP ØDS

this will change the baud rate of the printer (port 1) to 300 baud and change the data format of the serial output (port 2) to 8 data bits and 1 stop bit. If the program is executed without any command line arguments, a menu will be displayed of the options that are available.

CONFIGIO

The CONFIGIO program allows the custom installation of the CP/AM operating system environment. The CP/AM environment areas that the CONFIGIO program can change are the hardware and software

terminal emulations, the keyboard key re-definitions table, and all of the systems input/output (console,list,... .etc) routines through special user installable I/O drivers. Note that all displayed data is in hex notation and the inputed data must also be in hex notation.

After running the program CONFIGIO, there will be a menu of five options. The first is that of the custom terminal emulation. After selecting option one, the screen will display the current software and hardware terminal screen function definitions. Under the software colunm for each of the screen functions will be the character sequences that a program will need to send to the system to accomplish that particular screen function. The exceptions to this are the lead-in, xy coord offset, and the xy xmit order.

The lead-in row is the lead-in character definition for any of the software screen functions that require a lead-in character. The xy coord offset and xy xmit order are the definition on how the address cursor information should be sent to the system. The hardware column shows the character sequence that the operation system will send out to the physical terminal when it receives one of the screen function requests from the software. Through this configuration most any program and terminal conbination can be accommodated.

The CP/AM operating system is sent out configured to emulate SOROC IQ 120/IQ 140 terminal function codes. With the CONFIGIO program, this can be changed at any time and can be changed back by selecting the default SOROC selection. Note that if the lead-in character is set to zero then the lead-in character will be disabled, also if any of the screen function definitions are zero they will be disabled.

The keyboard redefinition option from the main menu will allow any key on the keyboard to be redefined to another key. There can be a maximum of six entries in the keyboard redefinition table. The system is sent out with several keys redefined for use with the Apple][+ computers. The Apple][+ computers lacked some keys (especially the left bracket key) that are used by CP/M programs. If CP/AM is to be operating in a Apple //e or //c, then it is advised that all of the entries in the keyboard redefinition table be deleted since the //e and //c have these keys. This can be accomplished by selecting the delete option in the keyboard redefinition menu and entering the hex notation for the key that was redefined to delete the redefinition.

Option number three on the main menu will allow a software input/output driver to be loaded in and installed into the CP/AM system. The ability to install a driver into the system is intended for non-standard hardware products that the CP/AM operating system does not support. If needed, these drivers may be supplied by the hardware manufaturer, if not, the user will need to write one. The format of the driver file is given later in the manual.

All of the preceding options only modified the operation system
in memory. This means if the system is rebooted all of the
previous changes would be gone and the old setting would return.
To make the current operation system configuration permanent it
must be written out to a system disk. Option four from the main
menu will allow system configurations to be read and written to
from a system disk. If the disk does not contain the system,
an error message will be displayed. The write option will write
out the current system configuration to a system disk. Any time
that the CP/AM system is booted from that disk it will come up
with the configuration that was last written to it. There is
also a read option. This will read in a system configuration from
another disk that may be different from the current system
configuration on the disk that is being used without needing to
permanently modify the disk currently being used.

BATCH PROCESSING

SUBREN

The SUBREN command allows CP/AM commands to be stringed together
for automatic processing. SUBREN creates 'procedure' files for
subsequent processing by the CALL program. The general form of
the SUBREN command is:

        SUBREN d:filenane argl,arg2,...argn

When SUBREN is executed it will search the disk in drive d: for
a 'batch' file named filename.SUB where filename represents an
unambiguous filename. (An ambiguous filename is when the wild
card characters are used to match one or more files on a disk.
For a file to be unambiguous do not use the wild card characters,
if wild card characters are used the resulting filename must
match only one file on the disk.) SUBREN will then create the
procedure file xx$.SUB containing control statements from batch
file filename.SUB, with symbolic arguments $i in the batch file
replaced in the procedure file by literal arguments argi obtained
from the SUBREN command. For example, the batch file APROC.SUB
containing the text lines:

        DIR $1.*
        STAT $1.$2

would be transformed by the command

A>SUBREN APROC SLM PRN

into the procedure file AA$.SUB containing the commands

        DIR SLM.*
        STAT SLM.PRN

26

The command

A>CALL A

may then be used to invoke this 'procedure A'. In this example,
SUBREN transformed the batch file APROC.SUB into the procedure
file AA$.SUB called procedure A.   The name of the procedure is
determined by the first character of the batch filename.   For
example, the batch file BPROC.SUB would be transformed by SUBREN
into the procedure file BB$.SUB, called procedure B. A procedure
file can have the name of any valid filename character with any
number of them may existing at the same time, being be
conditionally or unconditionally invoked by CALL.

If double dollar signs '$$' appear in a batch file they will be
translated by SUBREN into a single dollar sign '$' in the
procedure file.   An alphabetic character 'x' preceded by the
up-arrow character will be translated by SUBREN into a single
control-x character in the procedure file. These conversions
allow any character that can be entered at the keyboard to be
placed into a procedure file.

CALL

The CALL program expands the scope of batch command processing by
calling any of a number of separate command procedures. For
example:

A>CALL B

will invoke the procedure contained in the file BB$.SUB.   Each
procedure file is a batch command file created by SUBREN.   The
character x is the name of the procedure and can be any
character, except '$', that is permissible in an unambiguous
filename.   Any file that has the name '$?$.SUB' at warmboot will
be shifted into batch mode and the next CP/AM command will come
from the file found which bypasses the advantages provided by the
CALL program.

Using the SUBREN program and a text editor, create a batch file
named AA$.SUB with the following text lines:

        STAT   *.ASM
        CALL   B

then create a second batch file named BB$.SUB containing:

        DIR   *.DOC
        DIR
        DIR   *.COM

There are now two batch files on the disk named AA$.SUB and
BB$.SUB.   When procedure A is called with the console command:

```
A>CALL A
```

the CALL program will search for a file named AA$.SUB, and if found, will rename this file A.SUB, then warmboot. Existence of a file name beginning with '$' will cause CP/AM to search for a file satisfying the ambiguous prototype $?$.SUB. CP/AM will find A.SUB and will therefore shift into batch mode and take its next command from the procedure file found. In the example, this command will be

```
STAT  *.ASM
```

Following STAT processing, CP/AM is still in batch mode and will take its next command from A.SUB. This command will be

```
CALL  B
```

The CALL program first will search for a file satisfying the ambiguous prototype $?$.SUB, and when a file such as A.SUB is found, it will be renamed AA$.SUB. This deactivates the current command procedure with the deactivated file containing all remaining unprocessed commands. CALL then searches, in the present case, for a file named BB$.SUB, and if found, renames it to B.SUB, then warmboots. When CP/AM searches for $?$.SUB it will find B.SUB and so the next command will be taken from the procedure file found, and in the example the command will be

```
DIR  *.DOC
```

This process will continue until all the commands in B.SUB have been processed. Note that it does not return to the previous procedure A.

CONDITIONAL CALLS

A system error flag (RERF) is defined under CP/AM on page zero at location 003E hex. Any applications program can set or clear this flag. The operating system does not modify this flag. The SETERR program can also be used to set the system error flag to any value 0 through 9.

Under control of the system error flag, command procedures can be conditionally invoked by CALL. For example, if the system error flag is nonzero,

```
A>CALL  B  T
```

will invoke the procedure contained in the file BB$.SUB; otherwise, if the system error flag is zero, control will pass to the next command of the current procedure, or to the console if no procedure file exists.

For demonstration purposes, the program SETERR will be used to simulate the action of an applications program that sets the system error flag to indicate that subsequent processing should

follow an alternate procedure.

Using a text editor, create a main procedure in MAIN.SUB containing the text lines:

```
SETERR Ø
CALL A T
SETERR 1
CALL A T
DIR
```

Create an alternate procedure in ALT.SUB containing the text lines:

```
SAVE 1 POSTRUN.TMP
STAT POSTRUN.TMP
```

Use SUBREN on both of these files to create the batch files MM$.SUB and AA$.SUB. The main procedure can be invoked with the console command

A>CALL M

The first command of procedure M, SETERR Ø, simulates action of an applications program that clears the system error flag to zero. The next command in the main procedure, CALL A T, will therefore act as a no operation, allowing control to pass to the following command, SETERR 1. This command simulates action of an applications program that sets the system error flag. The next command, CALL A T, will invoke procedure A, the alternate procedure.

The CALL program can also use condition codes stored in the system error flag to conditionally invoke procedure files. Condition codes range from Ø to 9 and if the code in the CALL program is equal to the code in the system error flag the procedure file will be invoked. In the following example the procedure file 'C' will be invoked if the condition code the system error flag is 5:

A>CALL C 5

Note that the SETERR program can set the system error flag to any value from Ø to 9.

SUBMIT

The SUBMIT program combinds the functions of SUBREN and CALL programs together so that one command can both create a procedure file and invoke it at one time. When executed the SUBMIT program will create a procedure file in the same way that SUBREN does, including arguments, and give it the filename of $@$.SUB. When the SUBMIT program finishes running and a warmboot is performed the procedure file will be automaticly shifted into batch mode.

This will eliminate the need to use the CALL program to start the batch processing. In the example that follows a series of commands will be executed from the procedure file 'STARTUP.SUB' to automaticaly run the RAMDRIVE program and bring up WordStar.

Create the file 'STARTUP.SUB' with the following lines:

```
RAMDRIVE
C:
A:WS
```

then execute AUTORUN with the following command line:

A>AUTORUN SUBMIT STARTUP

The next time the disk is cold booted the WordStar will come up with the ramdisk as the default drive.

This concludes the introduction to the CP/AM system and supplied programs. For the general user the remaining sections of the manual can be skipped, but for those who have knowledge of 6502 and Z80 assembly language and plan to write CP/AM system programs push on.

6502/Z80 ADDRESS TRANSLATION

In order to get the most amount of contiguous memory for the CP/AM operating system and from the differences between how the Z80 and 6502 processors treat memory areas, it was necessary to reorganize the Z80 memory map in the Apple. This is accomplished through hardware on the Z80 card and is completely invisible to either processor. The effect of this re-mapping of the Z80 memory only becomes apparent when trying to execute 6502 subroutines from the Z80 card or when trying to address a specific Apple memory location by the Z80. The following chart will show the correspondence between memory accesses from the 6502 to that of the Z80.

Z80 ADDRESS	6502 ADDRESS
0000H-0FFFH	$1000-$1FFF
1000H-1FFFH	$2000-$2FFF
2000H-2FFFH	$3000-$3FFF
3000H-3FFFH	$4000-$4FFF
4000H-4FFFH	$5000-$5FFF
5000H-5FFFH	$6000-$6FFF
6000H-6FFFH	$7000-$7FFF
7000H-7FFFH	$8000-$8FFF
8000H-8FFFH	$9000-$9FFF
9000H-9FFFH	$A000-$AFFF
A000H-AFFFH	$B000-$BFFF
B000H-BFFFH	$D000-$DFFF
C000H-CFFFH	$E000-$EFFF
D000H-DFFFH	$F000-$FFFF
E000H-EFFFH	$C000-$CFFF
F000H-FFFFH	$0000-$0FFF

CALLING 6502 SUBROUTINES

To call a 6502 subroutine, the address of the routine to be called must be placed at memory location F3D0H. The address of subroutine must be given as a 6502 address and not Z80. The relation between the two addresses are given in the 6502/Z80 Address Translation Table. Information can be passed to the subroutine through the 6502 registers. When the subroutine is called the 6502 registers are automaticaly loaded from the memory locations shown below. When all of the needed information is initialized the subroutine can be called by performing a write operation to the slot address that the Z80 card is located in, which can be found in location F3DEH as shown below.

When the write operation is performed the Z80 card will be stopped where it is and 6502 will start running. The 6502 will

first load its registers from the locations shown below and then perform a jump to subroutine operation to the address stored at F3D0H. When the end of the subroutine is reached, by executing a return from subroutine instruction, the 6502 will store the contents of its registers back into the locations shown below, for the Z80 to examine, and perform a write operation to the slot address that the Z80 card is located in. This will stop the 6502 and startup the Z80 where it left off, ready to execute the next instruction.

Note: When the 6502 is activated, the monitor roms are enabled for the 6502 to use. Care should be taken for the language card ram is still write enabled during this time.

```
Z80 ADDR   6502 ADDR              PURPOSE

F045H      $0045                  6502 A register pass area
F046H      $0046                  6502 X register
F047H      $0047                  6502 Y register
F048H      $0048                  6502 P (status) register
F049H      $0049                  6502 S (stack) register


F3D0H      $03D0                  Address of 6502 subroutine to be called

F3DEH      $03DE                  Address of the slot the Z80 card is in
```

This is a sample Z80 call to a 6502 monitor subroutine to beep the Apple speaker:

```
;
;Subroutine to beep the Apple speaker
;
;     equates
;
Z80        EQU    0F3DEH          ;LOCATION OF Z80 CARD STORED HERE
A$SUB      EQU    0F3D0H          ;ADDRESS OF 6502 SUBROUTINE TO CALL
                                  ; GOES HERE
REG$A      EQU    0F045H          ;6502 A REGISTER GOES HERE
;
BEEP       EQU    0FF3AH          ;6502 ADDRESS OF SUBROUTINE TO CALL
;
START:     XOR    A               ;CLEAR A REGISER
           LD     (REG$A),A       ;IN THIS CALL IT SERVES NO PURPOSE
                                  ; OTHER THAT TO DEMONSTATE PARAMETER
                                  ; PASSING
           LD     HL,BEEP         ;LOAD HL WITH ADDRESS TO CALL
           LD     (A$SUB),HL      ;STORE 6502 ADDRESS FOR 6502 TO USE
           LD     HL,(Z80)        ;GET ADDRESS OF Z80 CARD SLOT
           LD     (HL),A          ;WRITE TO SLOT
;
; THE 6502 EXECUTES SUBROUTINE HERE AND RE-ACTIVATES THE Z80
;
           RET                    ;RETURN FROM BEEP CALL
```

CP/AM SYSTEM TABLES

There are several tables in the CP/AM operating system that programs can access to find out information about the system environment. These tables are listed below:

This is the internal format of the terminal redefinition table. Refer to the CONFIGIO program desciption for more on how the software to hardware transfer works.

SCREEN FUNCTION DEFINITION TABLE

SOFTWARE	HARDWARE	DESCRIPTION
F396H	F3A1H	Cursor coordinate offset is contained in the lower 7 bits of this byte. The high bit is the transmit order of the X and Y coordinates. If the high bit is zero then the Y coordinate is transmitted first, if one the X coordinate is first.
F397H	F3A2H	Lead-in character, zero if no lead-in.

Note: In following bytes the high bit signals if a lead-in character is needed, the lower 7 bits contain the character that corresponds to that particular function. If the byte for a function is zero then that function is disabled.

F398H	F3A3H	Clear screen
F399H	F3A4H	Clear to end of page
F39AH	F3A5H	Clear to end of line
F39BH	F3A6H	Normal text
F39CH	F3A7H	Inverse text of high-line
F39DH	F3A8H	Home cursor
F39EH	F3A9H	Address cursor
F39FH	F3AAH	Move up one line
F3A0H	F3ABH	Non-destructively move cursor forward

REDEFINITION TABLE

The keyboard redefinition table can contain up to six key redefinitions. Each key from the keyboard will be checked against this table for a match. If a match is found, then the character is replaced with its redefinition. If there is no match, then the character is passed on. If the table is not full, the end is marked by the high order bit in the first non-used entry being set to one.

KEYBOARD REDEFINITION TABLE

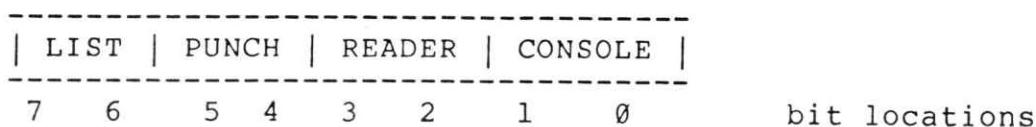|	CHARACTER	REDEFINITION
1.	F3ACH	F3ADH
2.	F3AEH	F3AFH
3.	F3BØH	F3B1H
4.	F3B2H	F3B3H
5.	F3B4H	F3B5H
6.	F3B6H	F3B7H

I/O DRIVER VECTOR TABLE

This table contains the vector addresses of the subroutines that perform all of CP/AM's I/O processes, excluding disk I/O. These vectors can be replaced with the CONFIGIO program to point to user subroutines. In this way, custom drivers can be installed into the CP/AM system to accommadate non-standard cards.

I/O VECTOR TABLE

VECTOR NUMBER	VECTOR NAME	DESCRIPTION
1	Console Status	Returns ØFFH in register A if a character is ready to read and ØØH if otherwise.
2	Console Input #1	Read a character from the
3	Console Input #2	console into the A register.
4	Console Output #1	Sends a character in
5	Console Output #2	register C to the console device. Register B contains zero if a standard character or if sending out a screen function command it contains the screen function number.
6	Reader Input #1	Reads a character from
7	Reader Input #2	the reader device into register A.
8	Punch Output #1	Sends the character in
9	Punch Output #2	register C to the punch device.
10	List Output #1	Sends the character in
11	List Output #2	register C to the printer device.

LOGICAL TO PHYSICAL I/O DEVICES

The CP/AM IOBYTE located at 0003H can be used to redirect the CP/AM logical input and output devices to different I/O vector table locations. The IOBYTE has the following structure:

```
-------------------------------------------------
|  LIST  |  PUNCH  |  READER  |  CONSOLE  |
-------------------------------------------------
  7   6    5   4     3   2      1    0            bit locations
```

Console field:

    0 - Console Input #1 and Console Output #1
    1 - Console Input #1 and Console Output #1
    2 - Reader Input #1 and List Output #1
    3 - Console Input #2 and Console Output #2

Reader field

    0 - Console Input #1
    1 - Console Input #1
    2 - Reader Input #1
    3 - Reader Input #2

Punch field

    0 - Console Output #1
    1 - Punch Output #1
    2 - Reader Input #2 (data is outputed through this vector)
    3 - Punch Output #2

List field

    0 - Console Output #1
    1 - Console Output #1
    2 - List Output #1
    3 - List Output #2

In order to use the CONFIGIO program to install custom drivers, a specially formatted file will have to be created. This file will be read in by option number three in the CONFIGIO program and will be automaticelly installed into the operating system. The format for this file is as follows:

    First byte: Number of patches to be made.

        Repeat for each I/O driver to install

    Next 2 bytes: Destination address of subroutine code.
    Next 2 bytes: Length of subroutine code.

    Next byte: Driver patch type 1 or 2

If driver patch type 1:

Next byte: I/O vector number to patch.
Next 2 bytes: Address to be placed into I/O vector
                        defined above.

            If driver patch type 2:

Next Byte: I/O vector number to patch.
Next 2 Bytes: Address into which to move the current
                        contents of I/O vector defined above.

Next 2 bytes: Address to be placed into I/O vector
                        defined above.

Next: The driver code is placed here after the
                        driver patch type information above.
                        The next byte after the driver code can
                        be another driver where the first byte
                        of the file defines how many driver I/O
                        patches to be made.

Inside the CP/AM system, there is a special area that is set
aside for user installed device drivers.   This area ranges from
F200H to F37FH.   If a driver is to be installed permanently into
the operating system, it must reside in this area too so that it
may be written out to a system disk with the   read/write I/O
configuration block option in the CONFIOGIO program.   Also note
that when the RAMDRIVE program is running the I/O block area
F280H to F37FH will be overwritten for the ramdrive program
usage.

Here is an example of a device driver patch to automatically
remove line feed characters for printers that automatically
advance to the next line after each carriage return.

```
;
;LIST DEVICE PATCH
;
        ORG     100H            ;THIS IS SO THE LOAD COMMAND WILL
                                ; LOAD IT AT 100H
ORIGIN  EQU     0F300H          ;REAL ORIGIN OF PROGRAM
OFFSET  EQU     ORIGIN-START
;
        DB      1               ;ONE PATCH TO BE MADE
        DW      0F300H          ;ADDRESS OF WHERE TO LOAD THE
                                ; DRIVER PATCH
        DW      PGMEND-START    ;LENGTH OF DRIVER CODE
;
        DB      2               ;PATCH TYPE
;
        DB      0AH             ;CHANGE I/O VECTOR 10
        DW      OUTPUT+OFFSET+1 ;ADDRESS WHERE TO PUT CURRENT VECTOR
                                ; CONTENTS
        DW      START+OFFSET    ;NEW ADDRESS TO PUT IN I/O VECTOR
```

```
;
;
START:      LD    A,C               ;MOVE OUTPUT CHARACTER TO A
            CP    0AH               ;CHECK FOR LINE FEED CHARACTER
OUTPUT:     CALL  NZ,0000H          ;SEND CHARACTER TO PRINTER IF NOT
                                    ; A LINE FEED, OTHERWISE CONTINUE
                                    ; ON WITHOUT PRINTING LINE FEED
            RET                     ;RETURN TO DRIVER CALLER
;
            END
```

The object code for this driver would be:

```
01 00 F3 07 00 02 0A 04 F3 00 F3 79 FE 0A C4 00 00 C9
```

PERIPHERAL CARD SLOT TABLE

This table contains information about what type of peripheral
cards are located in each slot of the Apple computer. The table
starts at location F3B8H, where F3B8H + S contains the ID byte
for each slot number S. Slot zero (F3B8H) does not contain a
card type ID byte but a count of the number of disk drive
controller cards found in the system. This counter increments by
two for each card found. Do not confuse this by thinking it
corresponds to the number of disk drives, for a system with only
one disk drive will have a count of two. The following table
describes the the meaning of each ID byte.

0 - No peripheral card ROM was found
1 - A peripheral card ROM was found, but not identified
2 - A disk controller card was found
3 - An Apple communications interface type card was found
4 - A high speed type serial card was found, usually an 80-colunm
    card
5 - A parallel printer type card was found
6 - A card following Apple Pascal 1.1 & PRODOS protocol

INTERRUPT HANDLING

From the way that the Z80 card uses the DMA line to deactivate
the 6502, all interrupt processing must be handled by the 6502.
To enable maskable interrupts, the IRQ solder pad on the Z80 card
must be connected, non-maskable interrupts are not available. If
the interrupt comes when the 6502 is active, interrupt handling
is done in the standard way. When the Z80 is active the
following steps must be followed:

    1. Save any registers that are destroyed on the stack
    2. Save the contents of the 6502 subroutine call address in
       case an interrupt has occurred during the preparation for
       a 6502 subroutine call
    3. Set up the 6502 subroutine call address to $FF58, which

is the address of a 6502 RTS instruction in the Apple monitor ROM.
4. Return control to the 6502 by writing to the slot address of the Z80 card. When the 6502 regains control it will immediately "see" the interrupt and jump to its interrupt handling routine. Upon return from the interrupt routine, the 6502 will execute the RTS instruction and reactivate the Z80 through the call manager.
5. When the Z80 is reactivated restore the previous 6502 subroutine call address
6. Restore all used Z80 registers from the stack
7. Enable Z80 interrupts with the EI instruction
8. Return from Z80 interrupt routine with RETI

WHY THE Z-80?

Your primary reason for having a Z-80 card in your Apple is so that you can run CP/M software in your computer. CP/M is an advanced DOS sold and marketed by various software producers under license by Digital Research, Inc. CP/M was the first microcomputer operating system. It is still the most widely supported one available. The wide support means more software choices for you as a user. You now have a choice of many sophisticated word processing, accounting, business and professional software packages when you have a Z-80 Plus card. Unlike standard Apple DOS, CP/M supports many languages in addition to Basic, Fortran, Cobol, and Basic Compiler are available. CP/M has many conveniences not found in Apple DOS such as fast disk I/O, better interface to machine language programs, simpler file transfer and wild card file naming conventions that allow you to refer to multiple files with one name.

WHAT IS THE PRIMARY DIFFERENCE BETWEEN THE Z-80 AND THE 6502?

The Z-80 is a register oriented processor whereas the 6502 is memory oriented. The Z-80 contains 18 eight bit registers and four 16 bit registers. Two accumulators and two flag registers are also provided. Moreover, all 8080 and 8085 assembly language programs will run unmodified on Z-80 systems because the 8080 and 8085 instruction code is a subset of the Z-80 instruction code. However, many instructions have been added eliminating the 8080's arithmetic and data processing shortcomings.

The CWS33X program

This program will allow WordStar 3.3x to work with the Apple //c.
CWS33X must be executed before WordStar can be used and will
remain in effect, thus allowing WordStar to work, until the next
time the CP/AM system is booted.

Fixing the inverse problem with WordStar 3.3x on Apple computer.

If the inverse video is backwards or not working properly, the
following fix can be used to correct the problem:

1.   Run the CONFIGIO program.
2.   Select option 1 on the main menu.
3.   Select option E on the terminal screen menu.
4.   Enter N for no leadin.
5.   Enter ØF for the character code.
6.   Enter H for hardware function.
7.   Select option F on the terminal screen menu.
8.   Enter N for no leadin.
9.   Enter ØE for the character code.
1Ø.   Enter H for hardware function.
11.   Enter X to exit to main menu.
12.   Select option 4 on the main menu.
13.   Enter W to write the new terminal configuration to the disk.
14.   Select X to exit the CONFIGIO program.

IN CASE OF DIFFICULTIES

I can't seem to get the cover off the IIc.

Well, it does come off. You may have to apply a little more force than you had originally thought. There are many plastic snaps that hold the cover down and some may be broken during the disassembly process; however, do not become overly concerned because the snaps are only used by Apple Computer to speed final assembly.

It's not clear which way the 65C02 plugs into the Z-80c.

With the Z-80c card installed in the IIc, the notch will point towards the volume control on the speaker.

When I install the Z-80c, it doesn't seem to lie down flat.

There are two large integrated circuits below the Z-80c. Sometimes Apple, when assembling the IIc, forgets to remove the chip carriers before these two chips are plugged into their sockets inside the IIc. The chip carriers are thin, plastic strips that are sandwiched in between the I.C. socket and the I.C. itself. These chip carriers cause the two large I.C.'s to not seat completely into their sockets. To remove the chip carriers, unplug the two I.C.'s, then pull out the chip carrier (NOT THE I.C. SOCKET ITSELF). Then replace the I.C.'s back in their socets making sure that their notches point toward the speaker volume control knob.

For Further Reading:

These books are available at most bookstores.

The CP/M Handbook by Rodney Zaks   *
Published by Sybex Inc.   a very good
2344 6th Street, Dept. A   CP/M manual
Berkeley, California 94710
1-800-227-2346

Introduction to Wordstar by Arthur Naiman   *
Published by Sybex   a very good
2344 6th Street,   Dept. A   WordStar manual
Berkeley, California 94710
1-800-227-2346

Microsoft BASIC   *
Published by Dilithium Press
11000 S.W. 11th Street Suite E
Beaverton, OR 97005
(503) 646-2713

dBASE II users guide
Pubished by Green/Prentice-Hall

CP/M and the Personal Computer
Dwyer and Critchfield/Addison-Wesley

Programming the Z-80 by Rodney Zaks
Published by Sybex Inc.
2344 6th Street, Dept A
Berkeley, California 94710

How to Get Started With CP/M by Carl Townsend
Published by Dilithium Press
P. O. Box 606
Beaverton, Oregon 97005

CP/M Assembly Language Programming by Ken Barbier
Published by Prentice-Hall, Inc.
Englewood Cliffs, New Jersey 07632

Z-80 Programming and Interfacing
Published by Howard Sams, Inc.
4300 West 62nd Street
Indianapolis, Indiana 46268

* Most recommended

Criticism of this manual is welcomed at all times. Welcomed are any comments which will enhance the content or format. Individuals that wish to contribute software are encouraged to do so.

Correspondence regarding this document should be addressed to:

APPLIED ENGINEERING
P.O. BOX 798
CARROLLTON, TX. 75006

Other Apple Peripherals by APPLIED ENGINEERING:

TIMEMASTER II H.O. Clock
8 Bit 8 Channel A/D
12 Bit 16 Channel 9 Gain A/D
16 Bit Input/Output Board

Z-80 Plus CP/M Card
RAMWORKS
Music Synthesizer
VIEWMASTER 80

and more on the way, much more............

APPLE PERIPHERALS ARE OUR ONLY BUSINESS
That's why we're so good at it!